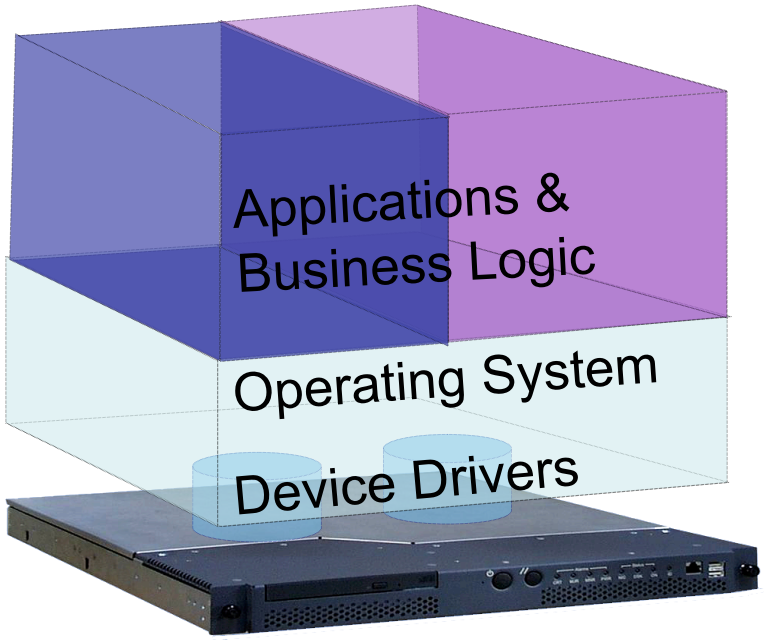
Xen and the Art of Multicore Processing

Simon Crosby, XenSource Inc

www.xensource.com

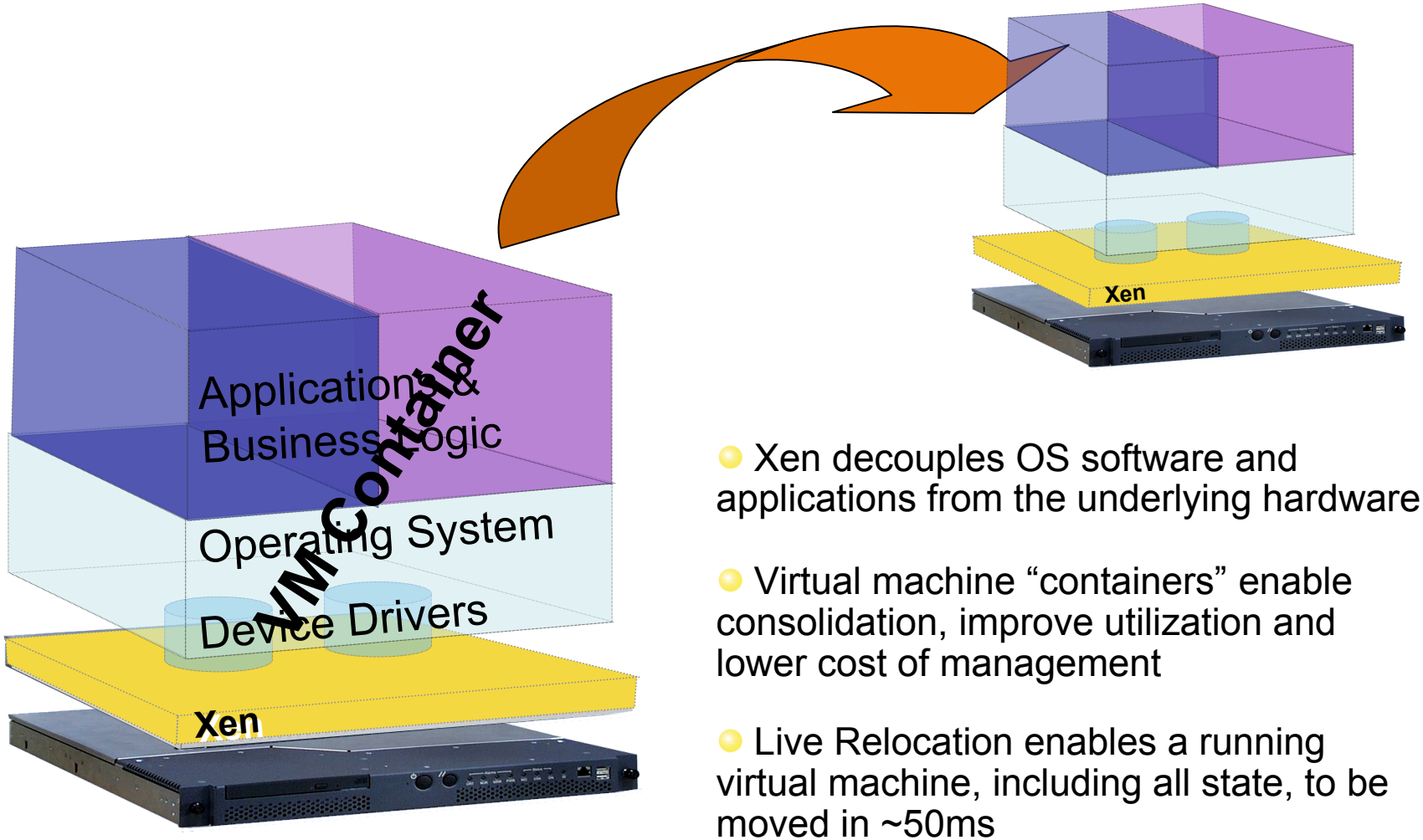
www.getxen.org

Problem: Success of Scale Out



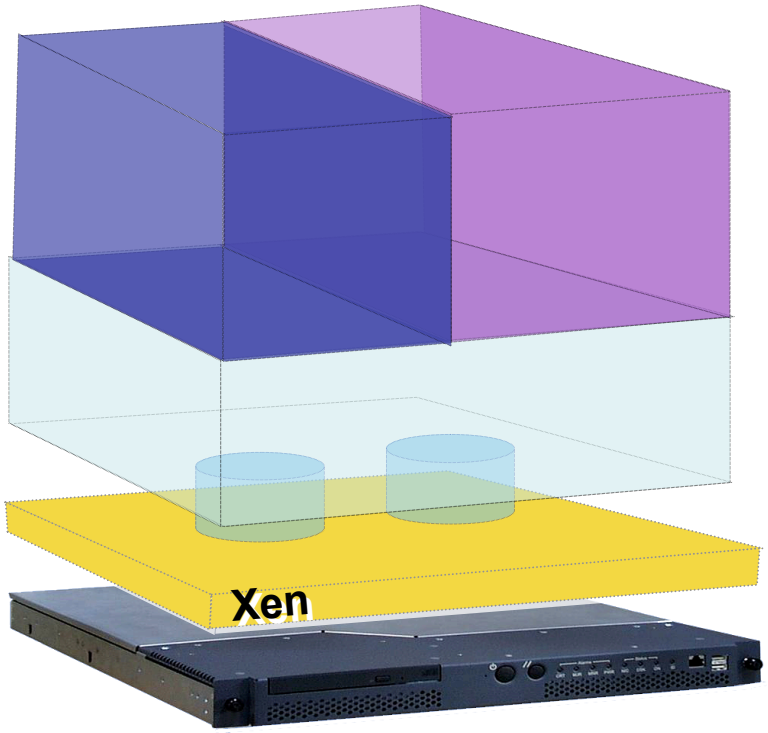
- One application per server - with resulting low utilization rates <15%
- Inability to scale resources for apps needing more resources during busy periods
- “OS+app” provisioning model takes a top-down view of the infrastructure
- Expensive to maintain, power, cool
- Inflexible

Along Came Xen



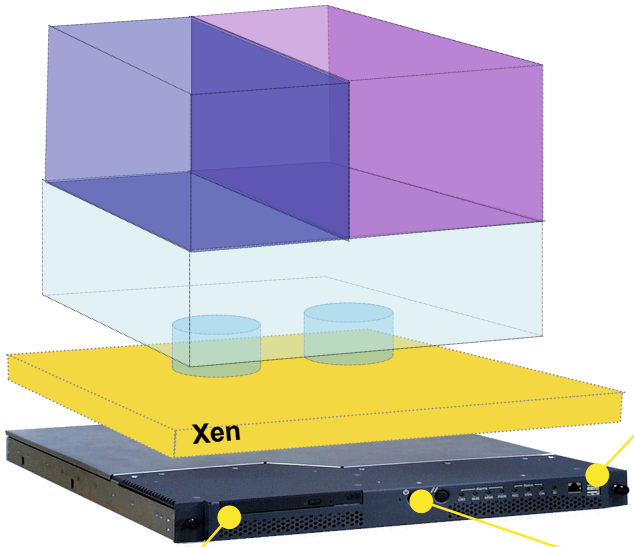
- Xen decouples OS software and applications from the underlying hardware
- Virtual machine “containers” enable consolidation, improve utilization and lower cost of management
- Live Relocation enables a running virtual machine, including all state, to be moved in ~50ms

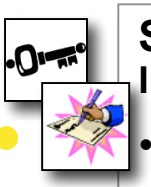
Xen Catalyzes Software Innovation



- Superb performance means it can go into heavy duty production
- Open Source means it can be universally improved and adopted
- Free, so it has a great opportunity for ubiquitous adoption
- Frees “stack $” for all players
- New opportunities for systems management, software distribution, licensing, fault management etc

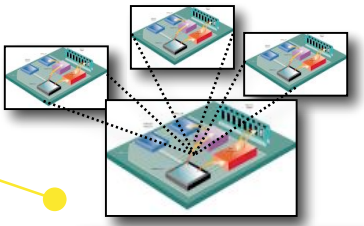
Xen Delivers Platform Innovation





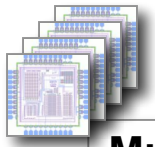
Security: Intel LT & AMT, AMD SEM, IBM TPM

- Building blocks for Trusted Computing infrastructure
- Enables delivery of Multi-Layer Secure systems
- Xen is the "secure platform base personality" dedicated to the CIO



Virtualization: Intel VT, AMD Pacifica, IBM Power

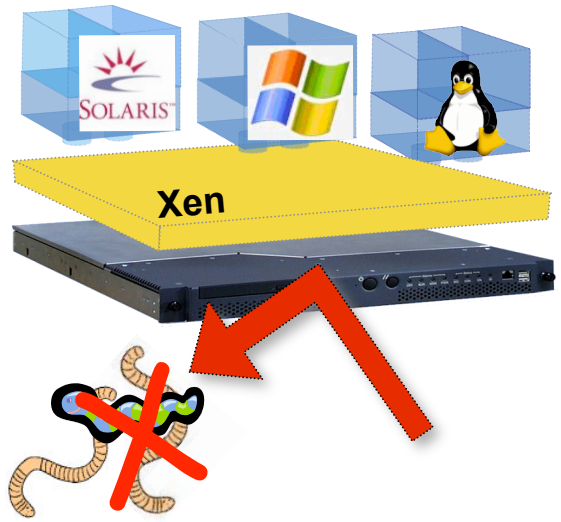
- CPU support accelerates virtualization
- Partitions system I/O so multiple OSes can share resources
- Xen gets leaner, more secure



Multi-core Processors & Hyper-threading

- Multiple parallel execution units capable of running SMP workloads
- Xen hides complexity from OS

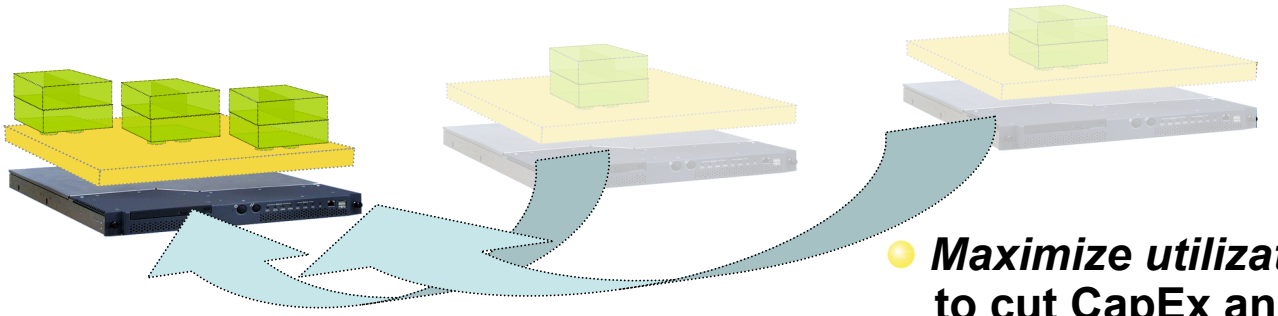
Xen Enhances Enterprise Security



Secure hypervisor layer "insulates" OSs and applications from attack

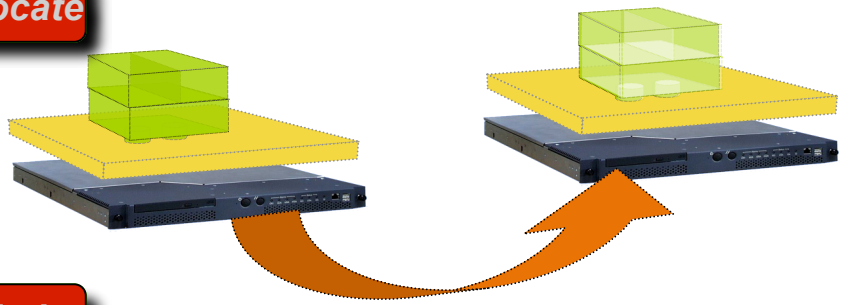
- DoS proof VMs due to superb resource partitioning
- Under 50K lines - independently scrutinized by the security community
- Foundation for Multi-Level-Secure Architecture (IBM, Intel, XenSource), leveraging new hardware security capabilities

User Benefits



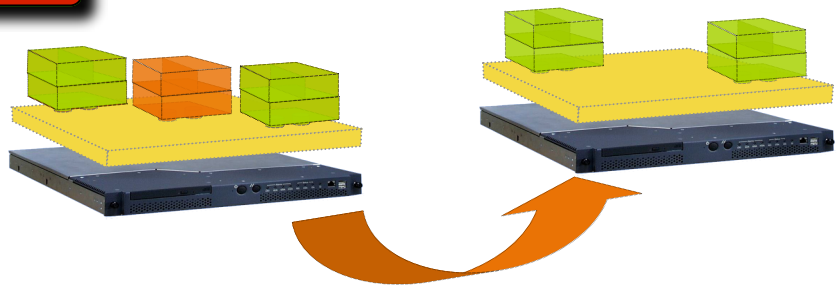
- ***Maximize utilization*** **of servers to cut CapEx and OpEx**

Relocate



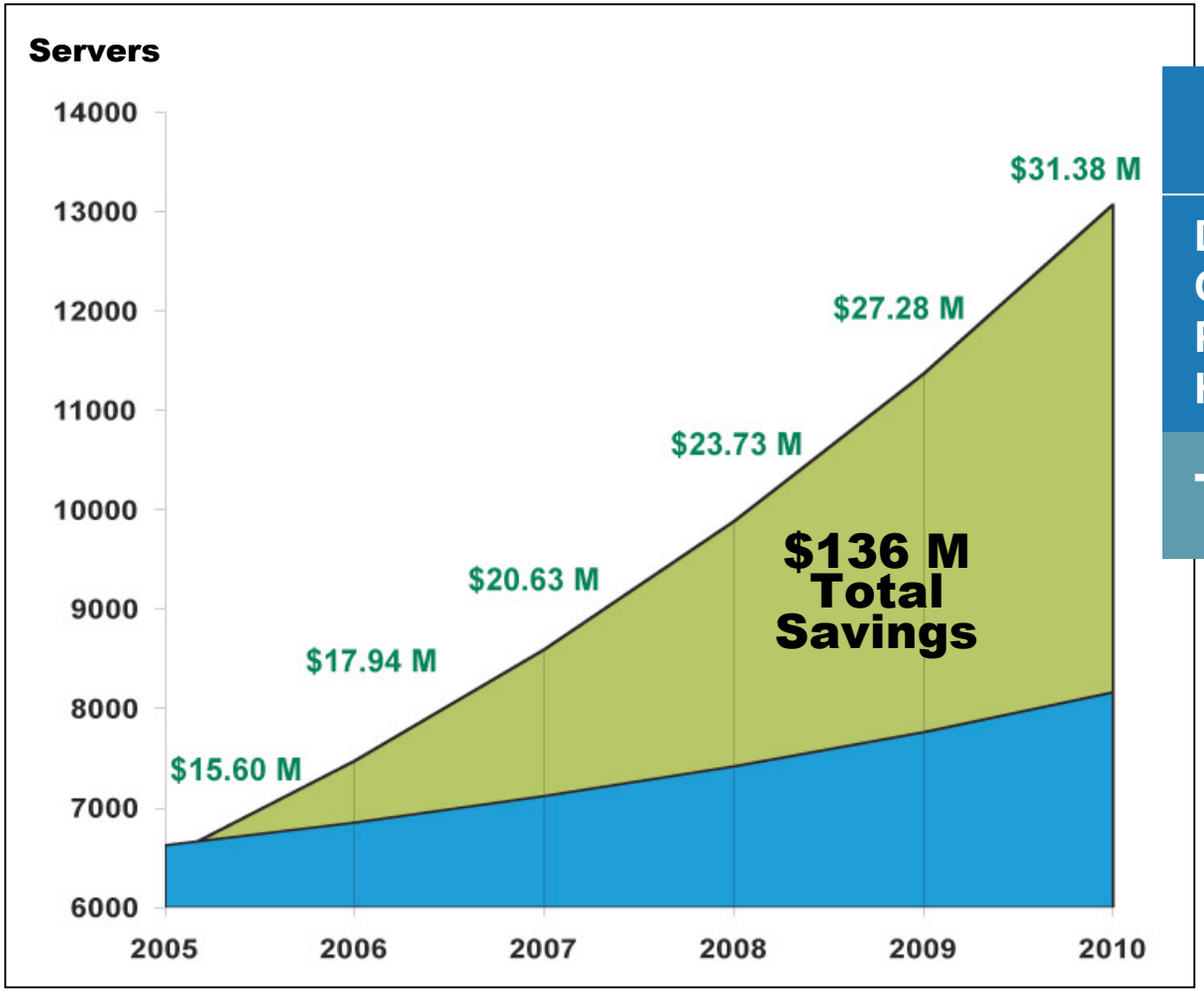
- **Relocate running VMs in ~50ms for maintenance, downtime etc**

Optimize



- **Dynamically *re-balance workload* to guarantee application SLAs**

Case Study: F100



SAVINGS BY AREA	
Deployment	$52M
Operators	$52M
Power/Cooling	$11M
H/W & S/W	$21M
Total	**$136M**

Adopted as An Industry Open Standard

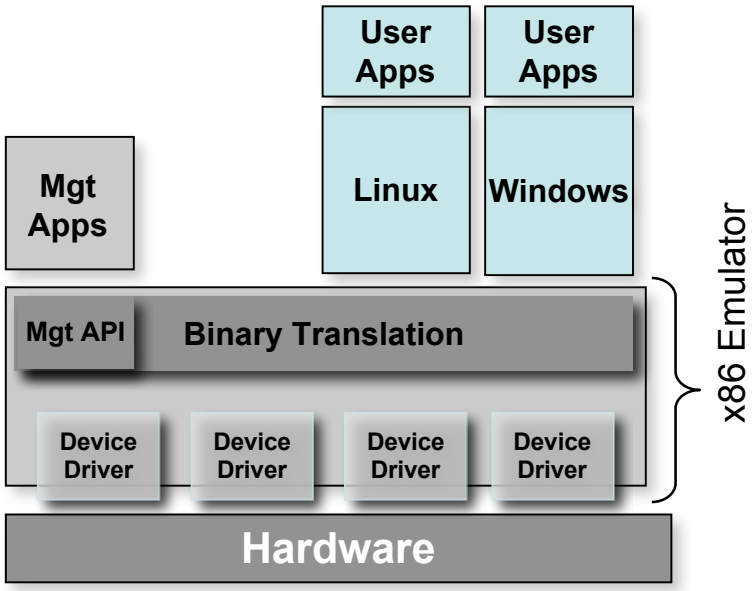
Operating System and Systems Management

Hardware Systems

Platforms & I/O

* Logos are registered trademarks of their owners

Virtualization Before Xen



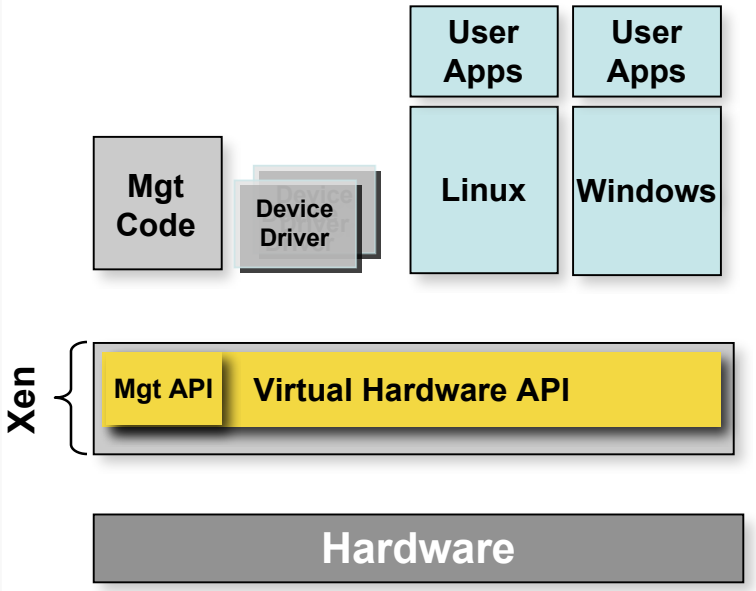
Existing Virtualization Products

- A (typcially proprietary) microkernel / OS under your OS
- Full virtualization requires binary patching of the OS at runtime
- Microkernel contains device drivers
- Emulates native chipset, so significant performance overhead
- Separate maintenance schedule for microkernel (& drivers) and the virtualized OS (& drivers)
- Vulnerable to driver failure
- Large code base
- But it runs unmodified OS images

Inside Xen, and Why it's so Cool

Xen: A Para-virtualizing Hypervisor

- Virtualizes (only) the base platform
  - CPU
  - MMU & Memory
  - Low level interrupts
- Small, reliable, efficient, trusted base-platform personality
- Guest OS co-operates with Xen
- Near native performance
- Lean (< 50KLOC) and getting leaner
- Supports native Linux device drivers
- Separates the driver from the guest
- No separate maintenance schedule
- Runs on x86_64, IA64, Power 5

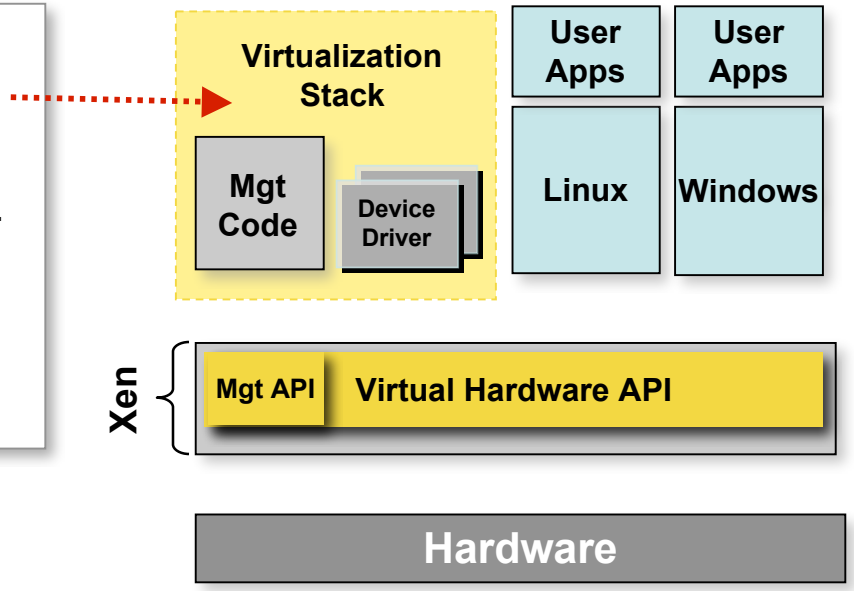


Free from your favorite Linux distro and www.getxen.org!

The Open Industry Virtualization Standard

Xen: Open for Innovation

- OEM specific drivers
- Management code: eg CIM agent, IPMI…
- Security policy
- Fabric specific support eg: Infiniband, SAN support

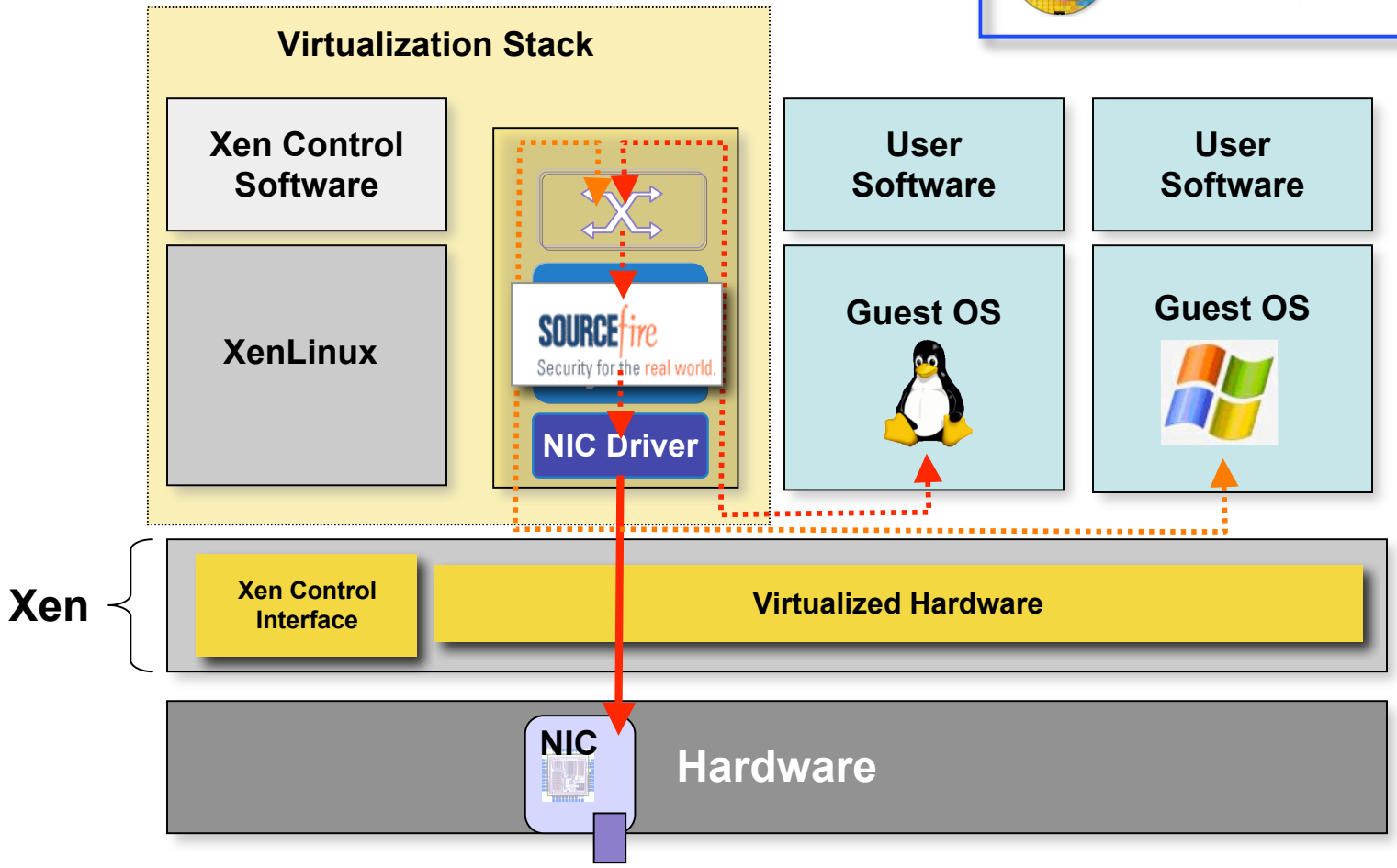


Xen provides a unique opportunity to ecosystem vendors to deliver value-added differentiation for

- –Manageability
- –Security
- –Device support

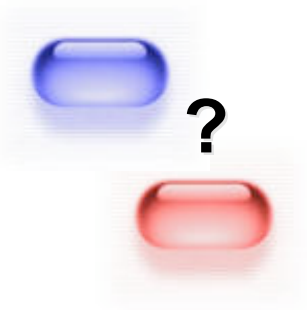
Example: Secure Network I/O

Intel Developer FORUM Fall 05 Solution Showcase ✓



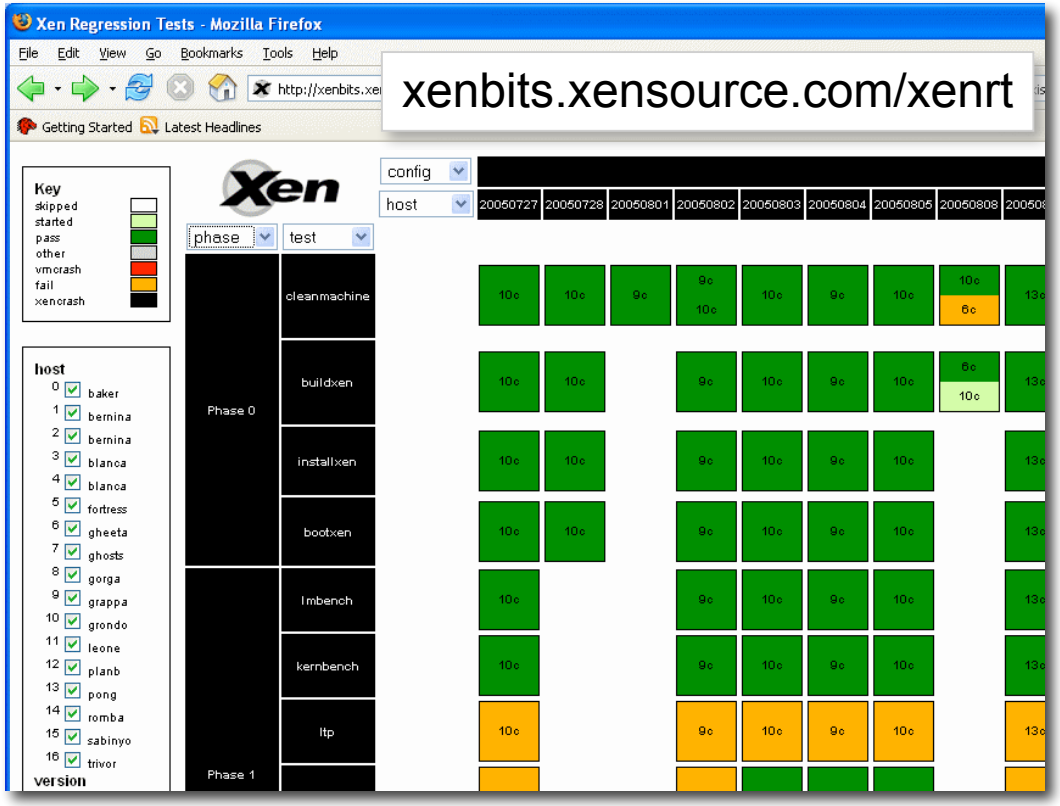
A Tribute to Open Source Process

- Xen is the industry's best performing hypervisor, secure, production ready
- 100% open source, developed by the best engineers at over 20 leading enterprises
- Great example of the benefit of open source
  - Ultimate escrow
  - Reduced risk for all players
  - Develops key features more quickly
  - Frictionless adoption with no proprietary agenda
- A common platform for innovation



Example: Xen Regression Testing

- Daily build & regression test for known corner cases & standard benchmarks
- Xen 3.0 Test CD tests all Xen hypervisor & guest combinations
- Tests functional components & system performance
- Results automatically uploaded to community web site - always available



XenRT is a powerful tool that allows the Xen project to benefit from the size & processes of major contributing vendors - everyone benefits

We're Hiring!

- Kernel engineers for Linux, Windows
- Customer support
- QA
- Product Management
- Product Marketing

In Palo Alto, CA and Cambridge, UK

Inside Xen - Platform Resource Virtualization

A Technology Deep Dive

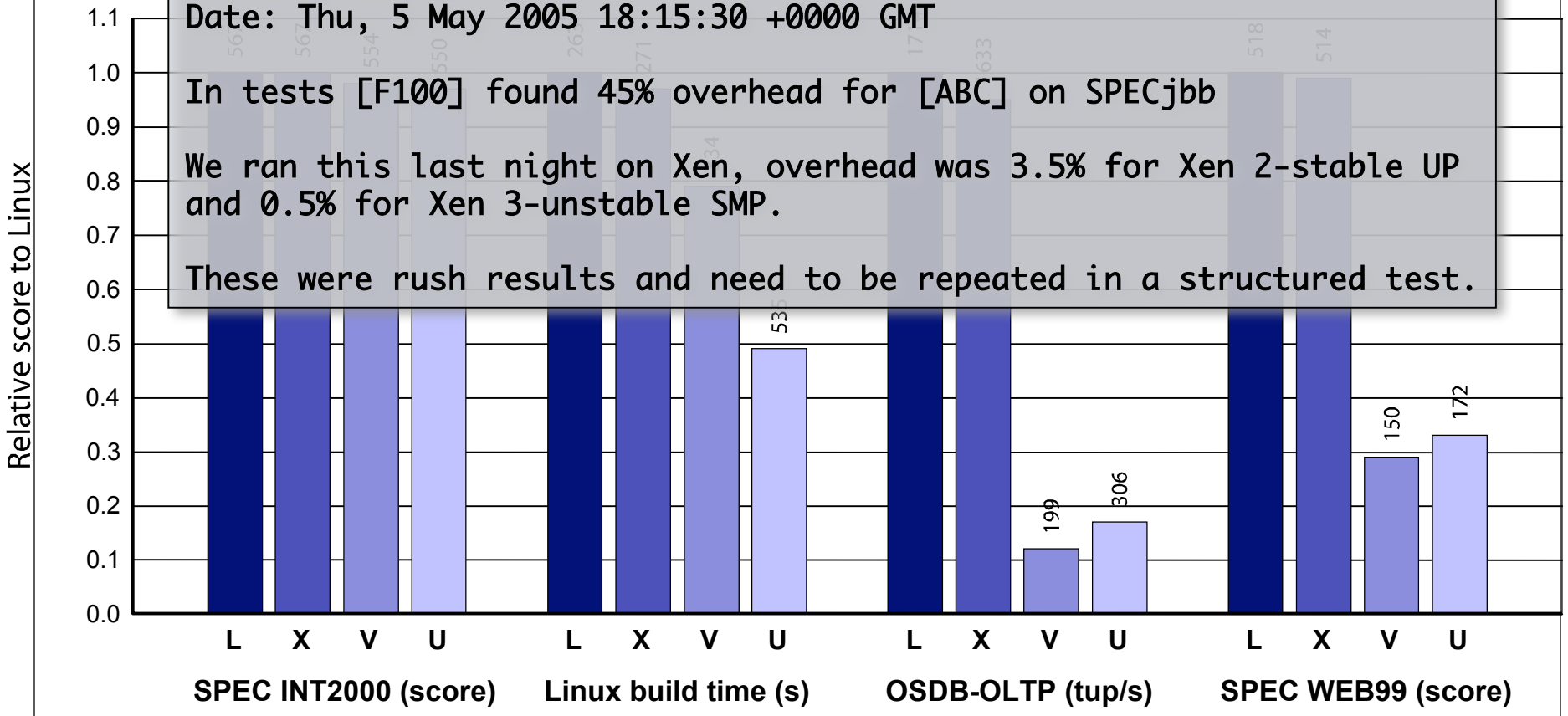
System Performance

From: rp@xensource.com
Date: Thu, 5 May 2005 18:15:30 +0000 GMT

In tests [F100] found 45% overhead for [ABC] on SPECjbb

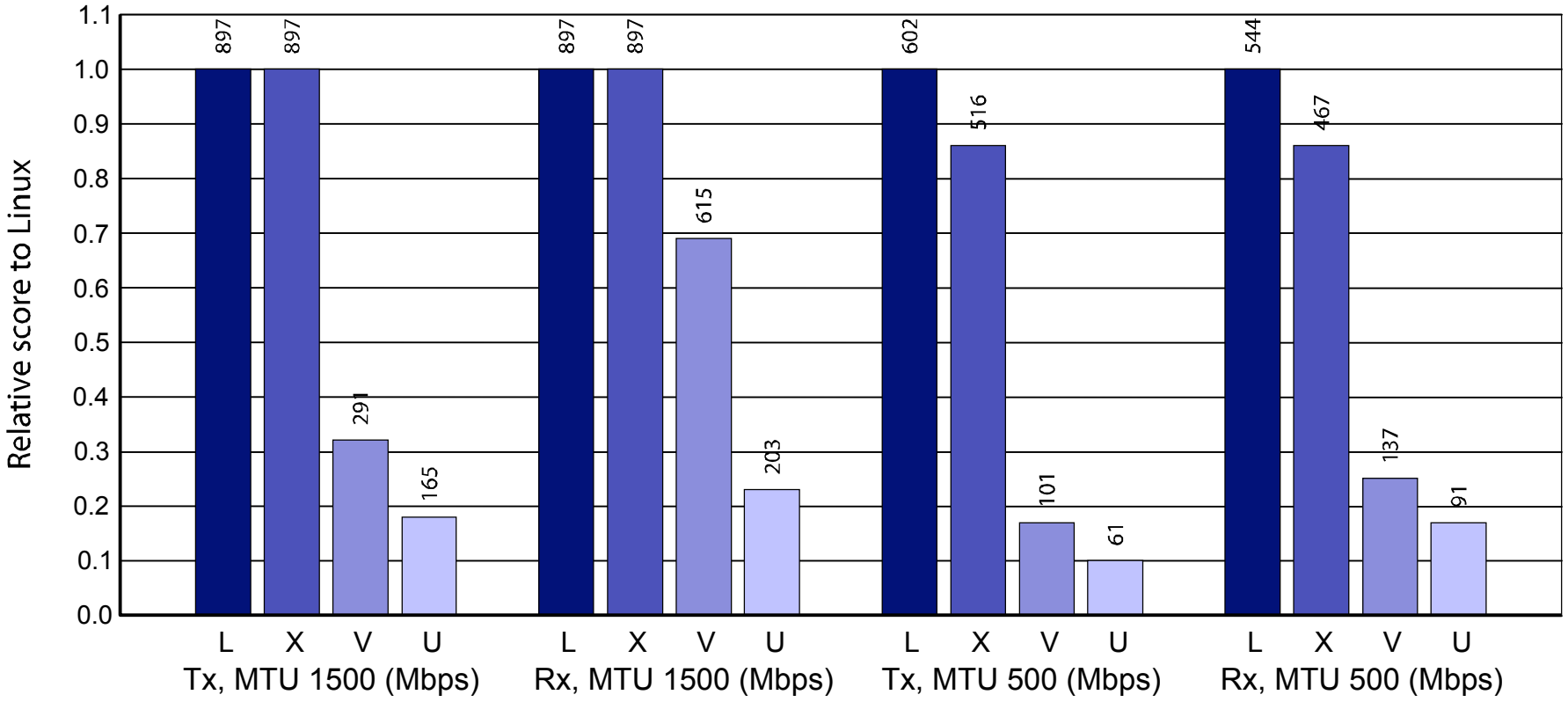
We ran this last night on Xen, overhead was 3.5% for Xen 2-stable UP and 0.5% for Xen 3-unstable SMP.

These were rush results and need to be repeated in a structured test.



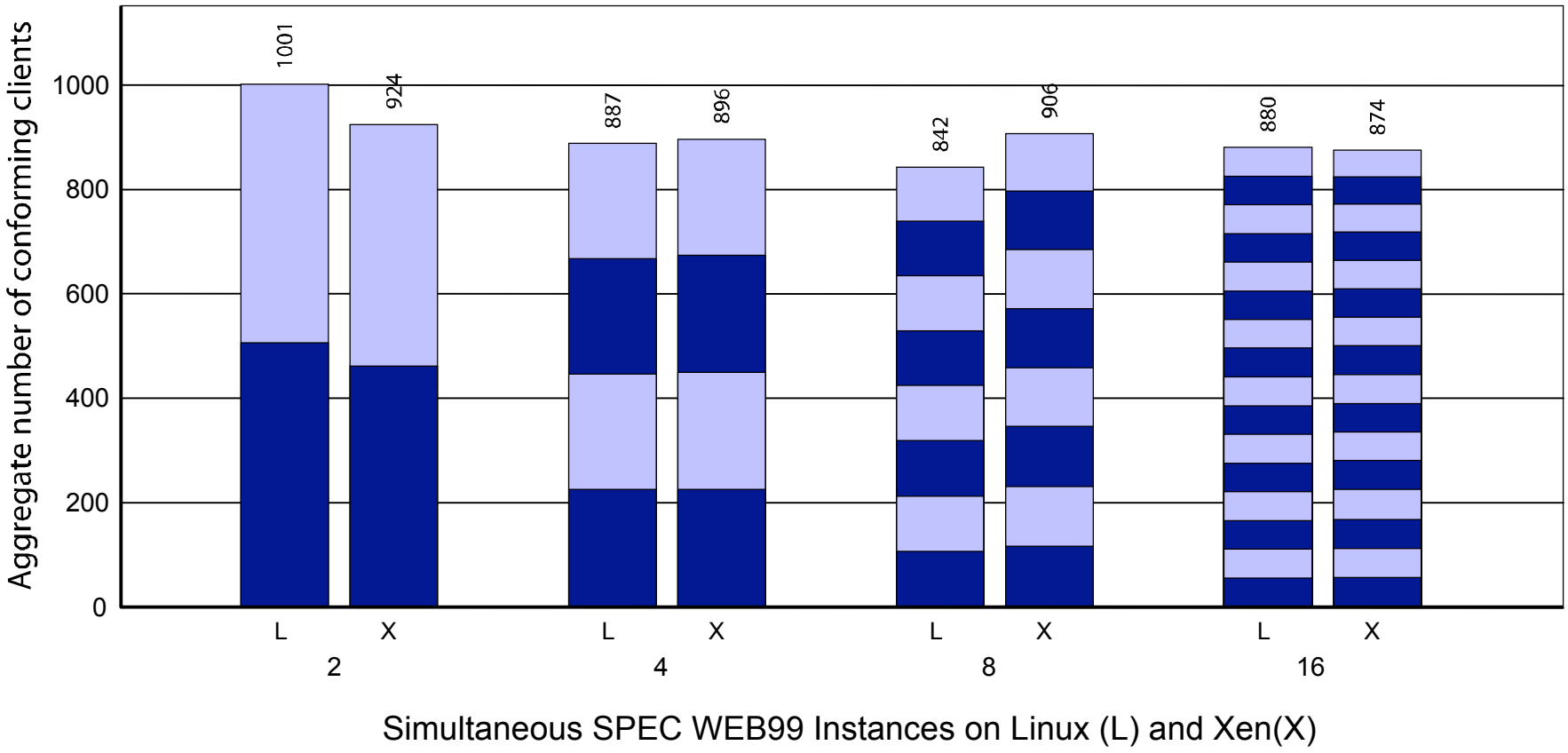
Benchmark suite running on Linux (L), Xen (X), VMware Workstation (V), and UML (U)

TCP results



TCP bandwidth on Linux (L), Xen (X), VMWare Workstation (V), and UML (U)

Scalability



Xen 3.0 Headline Features

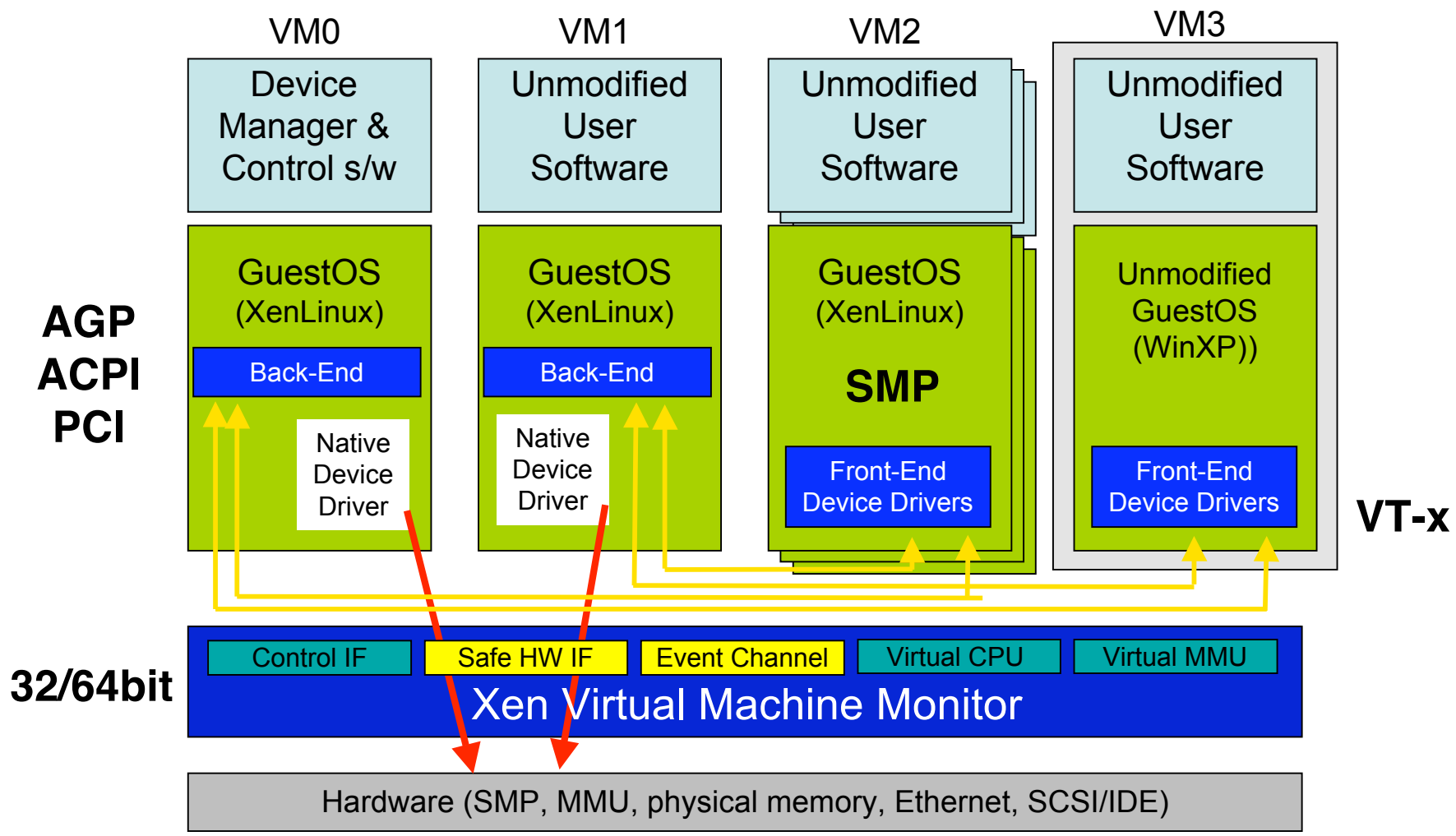
Target 3Q05 (in community testing)

- SMP guest OSes
- Supports unmodified guests on Intel VT processors
- Other extensions
  - x86/64: both AMD64 and EM64T
- Hardware compatibility: Graphics cards, ACPI, APM
- PAE support
- TPM 1.1 and 1.2 support; IBM sHype architecture

Other Projects Under Way:

- Para-virtualized Solaris x86 on Xen now booting (Sun)
- Xen / IA64 (HP)
- Xen / Power5 (IBM)
- Xen / SPARC (Sun)

Xen 3.0 Architecture



x86 CPU virtualization

- Xen runs in ring 0 (most privileged)
- Ring 1/2 for guest OS, 3 for user-space
  - GPF if guest attempts to use privileged instructions
- Xen lives in top 64MB of linear address space
  - Segmentation used to protect Xen as switching page tables too slow on standard x86
- Hypercalls jump to Xen in ring 0
- Guest OS may install ‘fast trap’ handler
  - Direct user-space to guest OS system calls
- MMU virtualisation: shadow vs. direct-mode

SMP Guest OSes

- Takes great care to get good performance while remaining secure
- Paravirtualized approach yields many important benefits
  - Avoids many virtual IPIs
  - Auto hot plug/unplug of CPUs (cores)
  - Key to dynamic addition / subtraction of resource
- SMP scheduling is a tricky problem
  - Strict gang scheduling leads to wasted cycles
- Xen 3.0 currently running on Unisys 32 way
  - (“Market leader” at 4 way)

SMP Guest Kernels

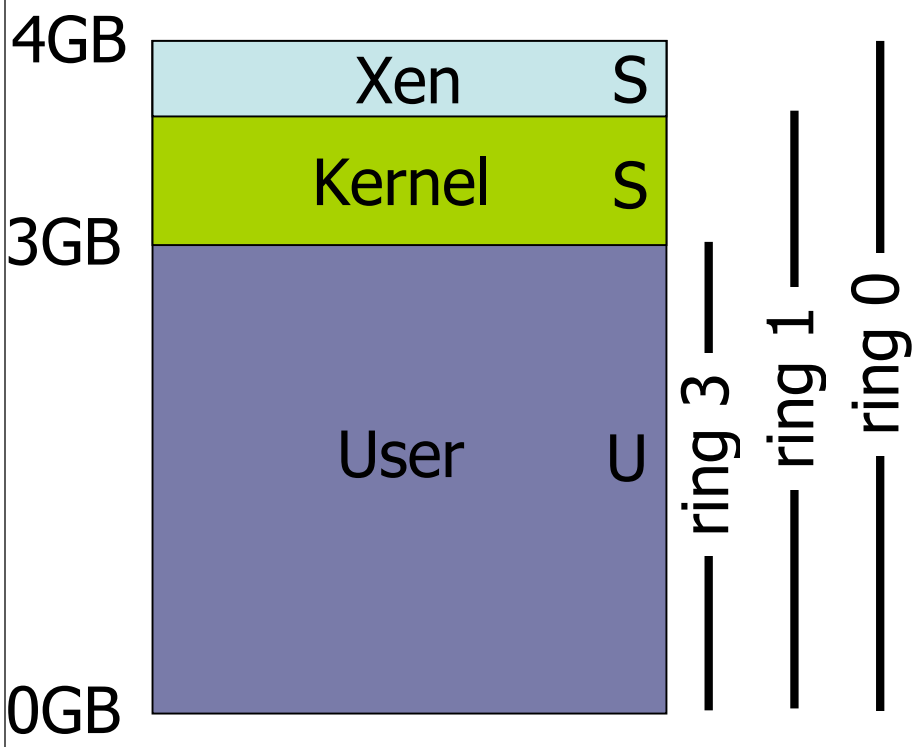
Xen extended to support multiple VCPUs

- Virtual IPI's sent via Xen event channels
- Currently up to 32 VCPUs supported

Simple hotplug/unplug of VCPUs

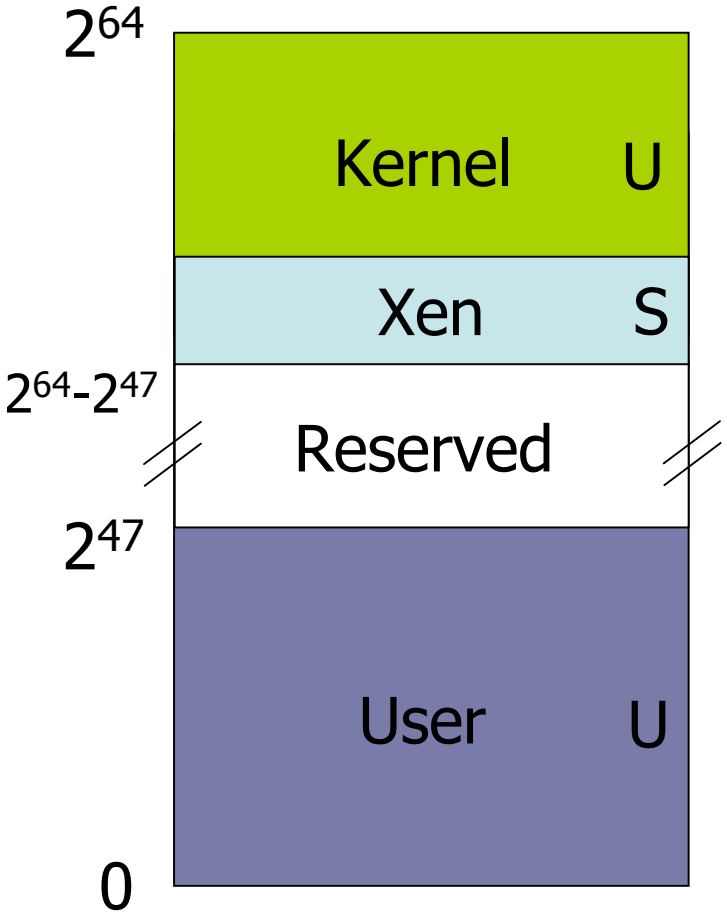
- From within VM or via control tools
- Optimize one active VCPU case by binary patching spinlocks

Protection: x86_32



- Xen reserves top of VA space
- Segmentation protects Xen from kernel
- System call speed unchanged
- Xen 3 now supports PAE for >4GB mem

Protection: x86_64

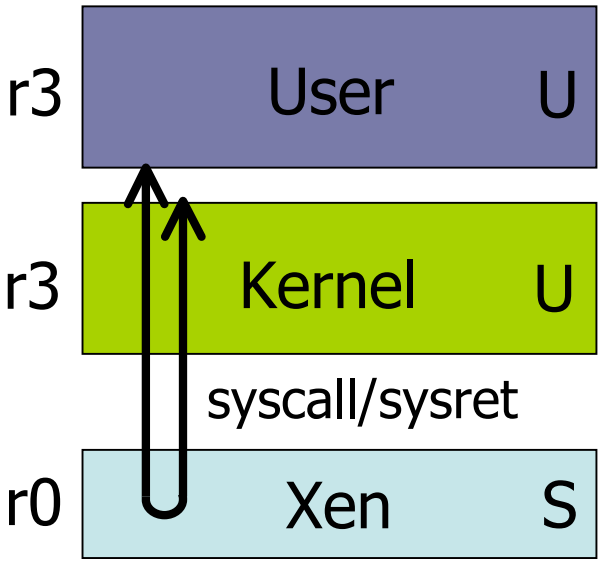


Large VA space makes life a lot easier, but:

No segment limit support

Need to use page-level protection to protect hypervisor

Protection: x86_64



Run user-space and kernel in ring 3 using different pagetables

- Two PGD's (PML4's), one with kern entries, one with guest+kern

System calls require an additional syscall/ret via Xen

Per-CPU trampoline to avoid needing GS in Xen

Para-Virtualizing the MMU

Guest OSes allocate and manage own PTs

- Hypercall to change PT base

Xen must validate PT updates before use

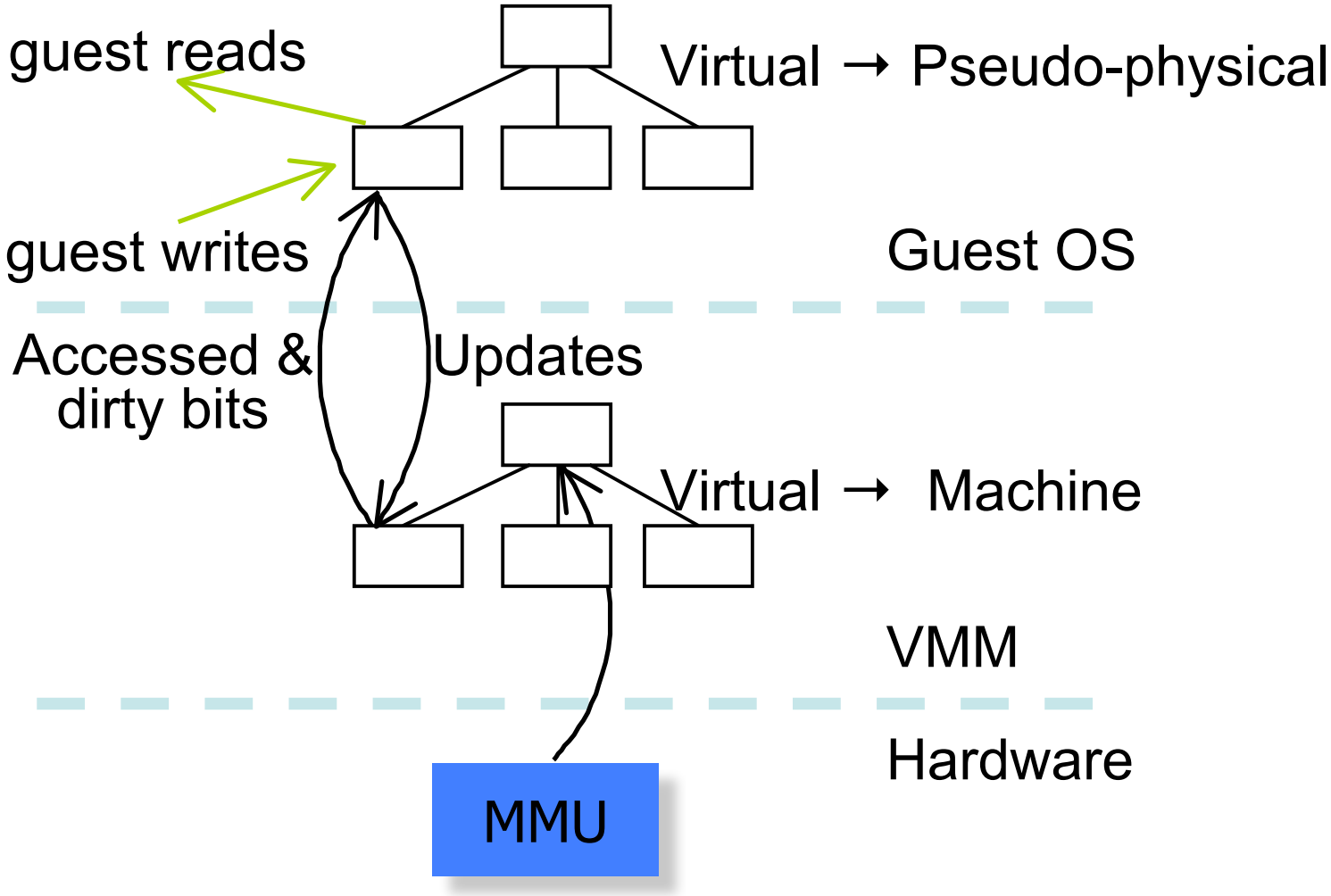
- Allows incremental updates, avoids revalidation

Validation rules applied to each PTE:

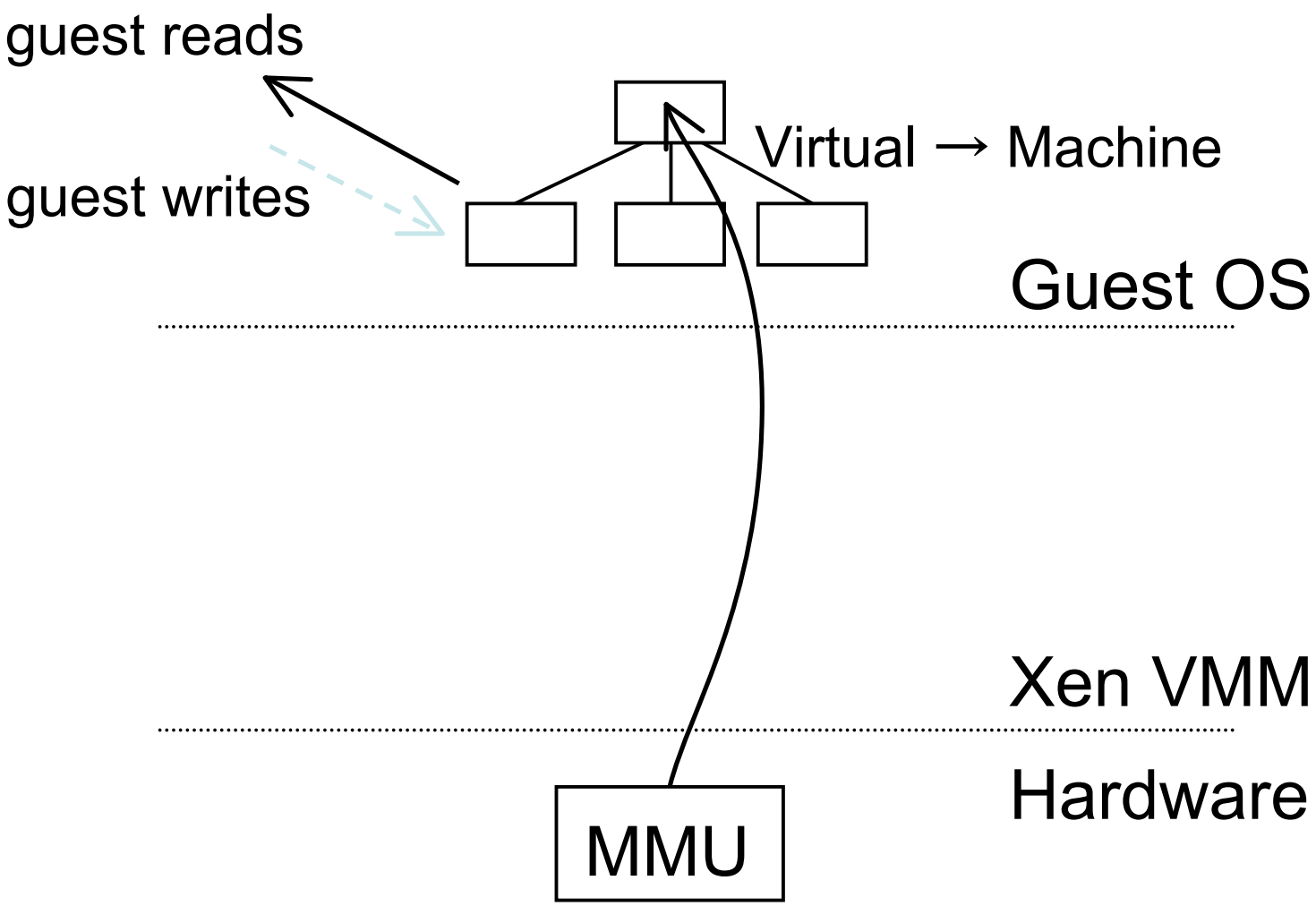
1. Guest may only map pages it owns*
2. Pagetable pages may only be mapped RO

Xen traps PTE updates and emulates, or 'unhooks' PTE page for bulk updates

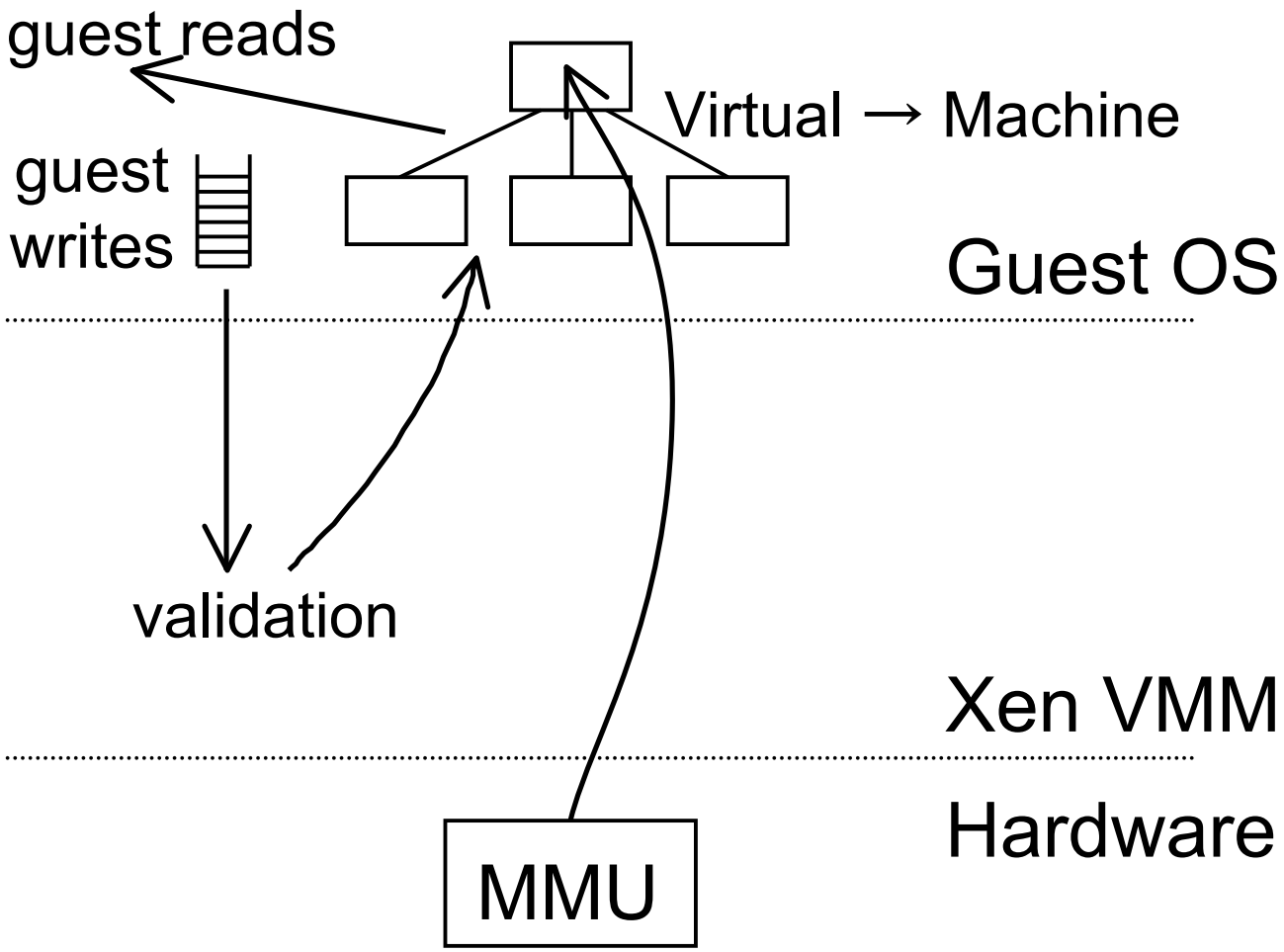
MMU Virtualizion : Shadow-Mode



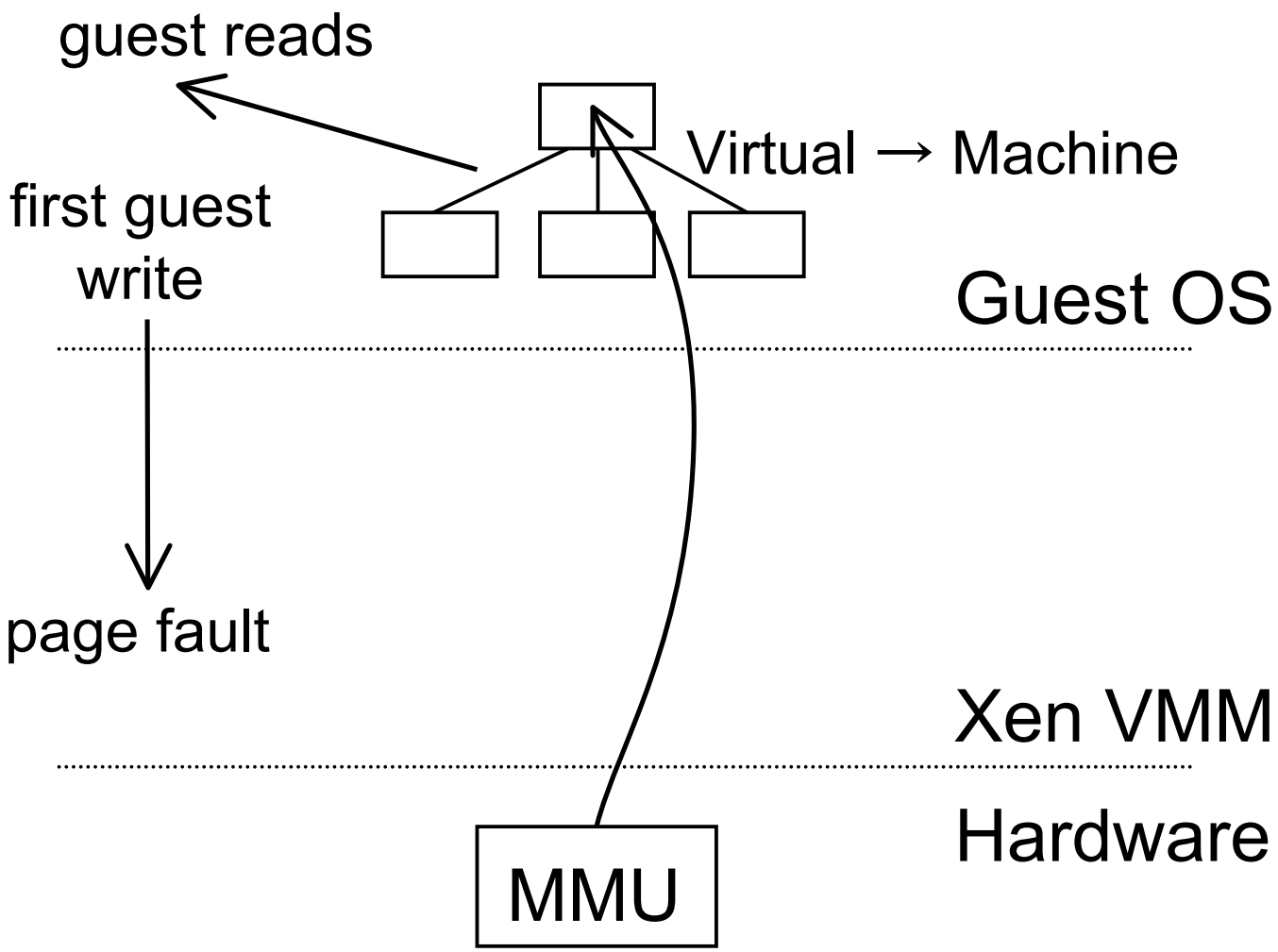
MMU Virtualization : Direct-Mode



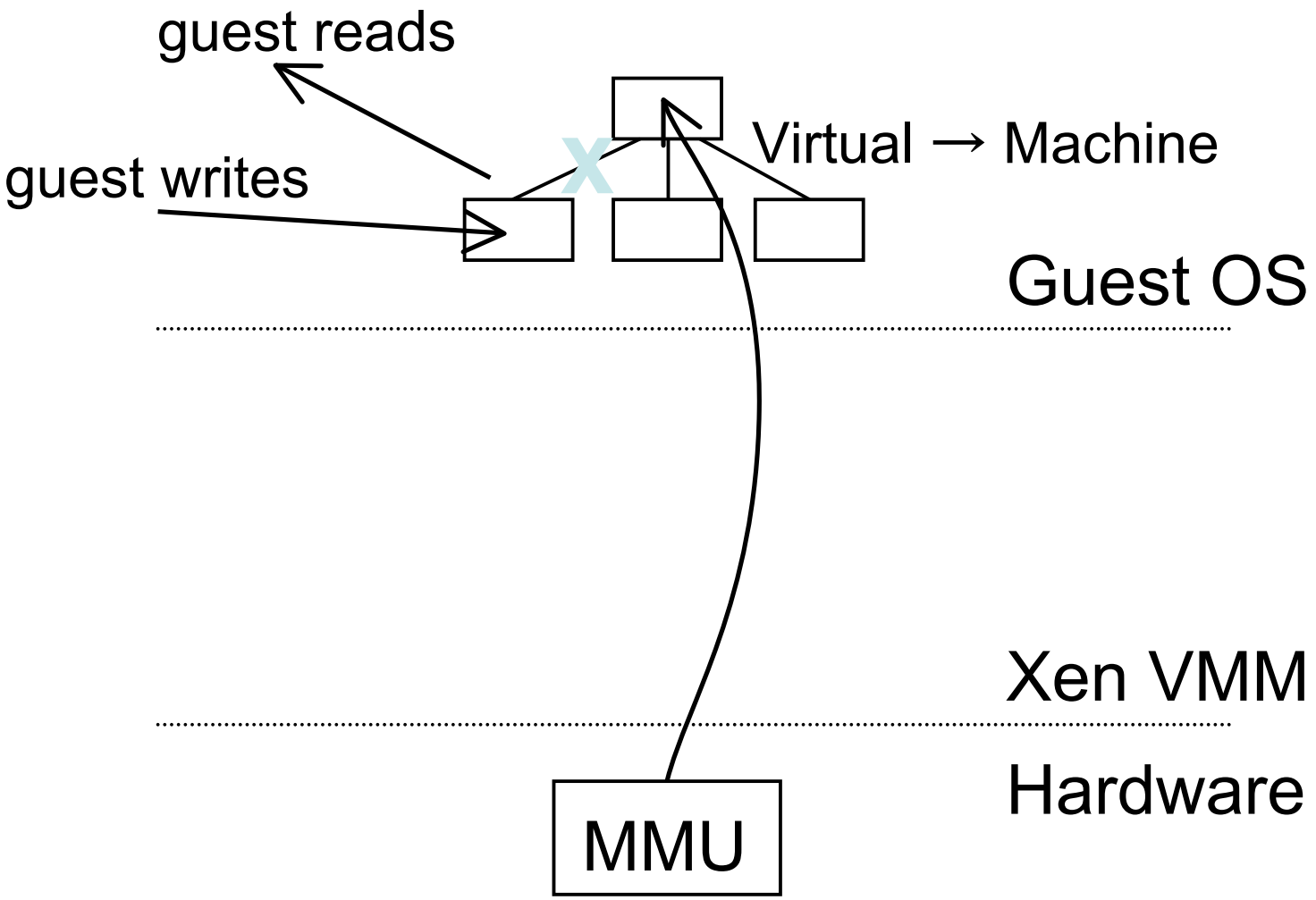
Queued Update Interface (Xen 1.2)



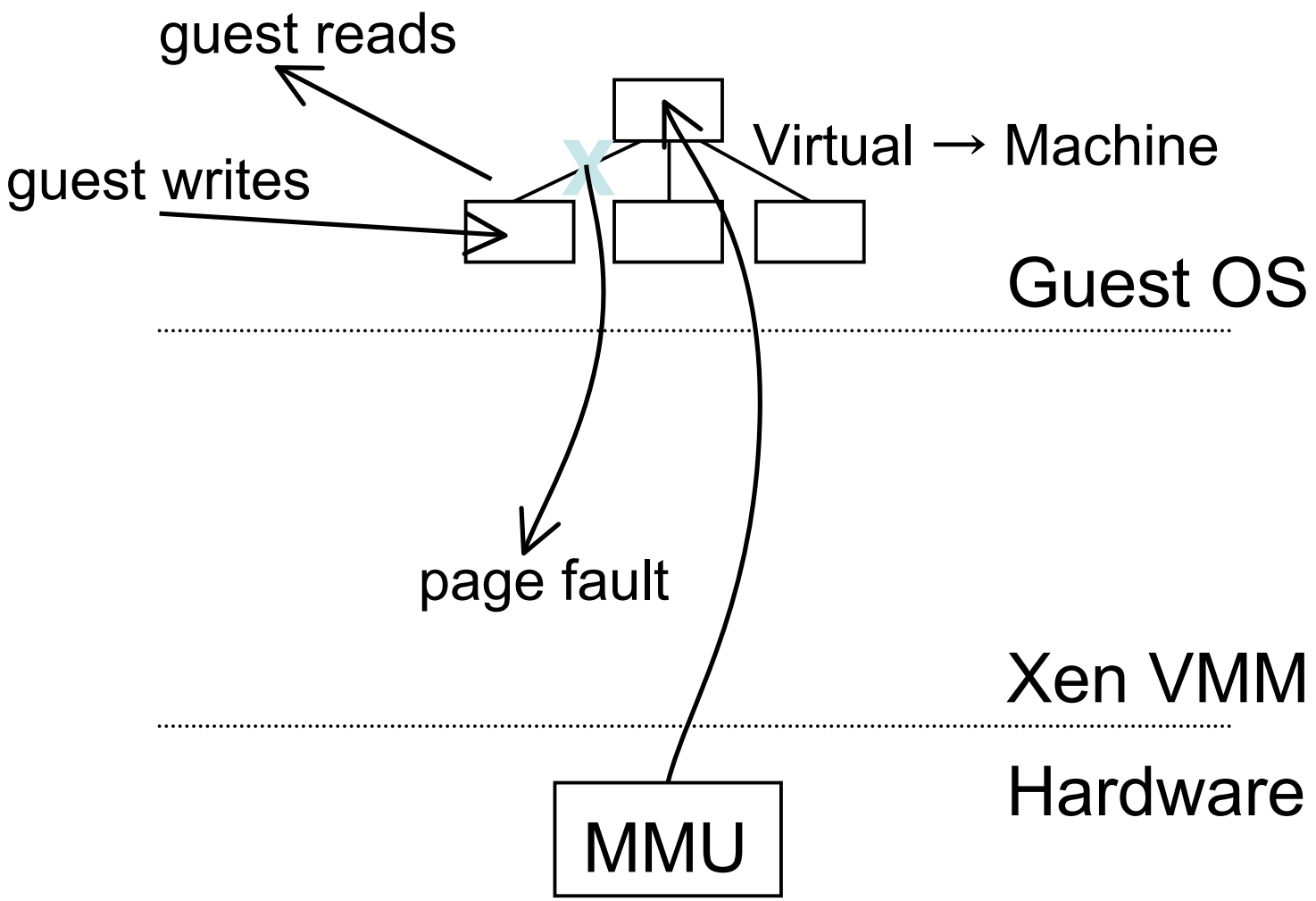
Writeable Page Tables : 1 – write fault



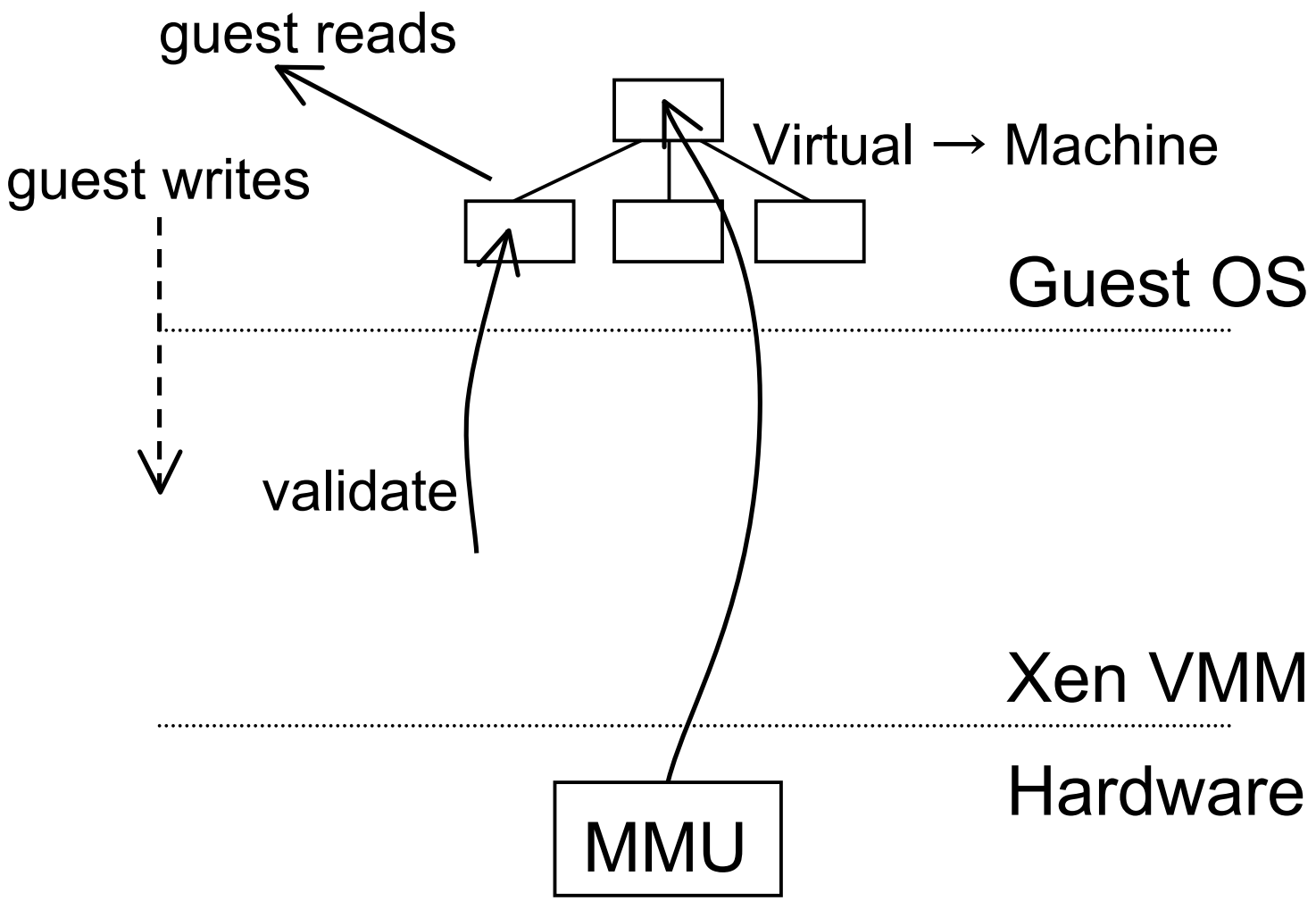
Writeable Page Tables : 2 - Unhook



Writeable Page Tables : 3 - First Use



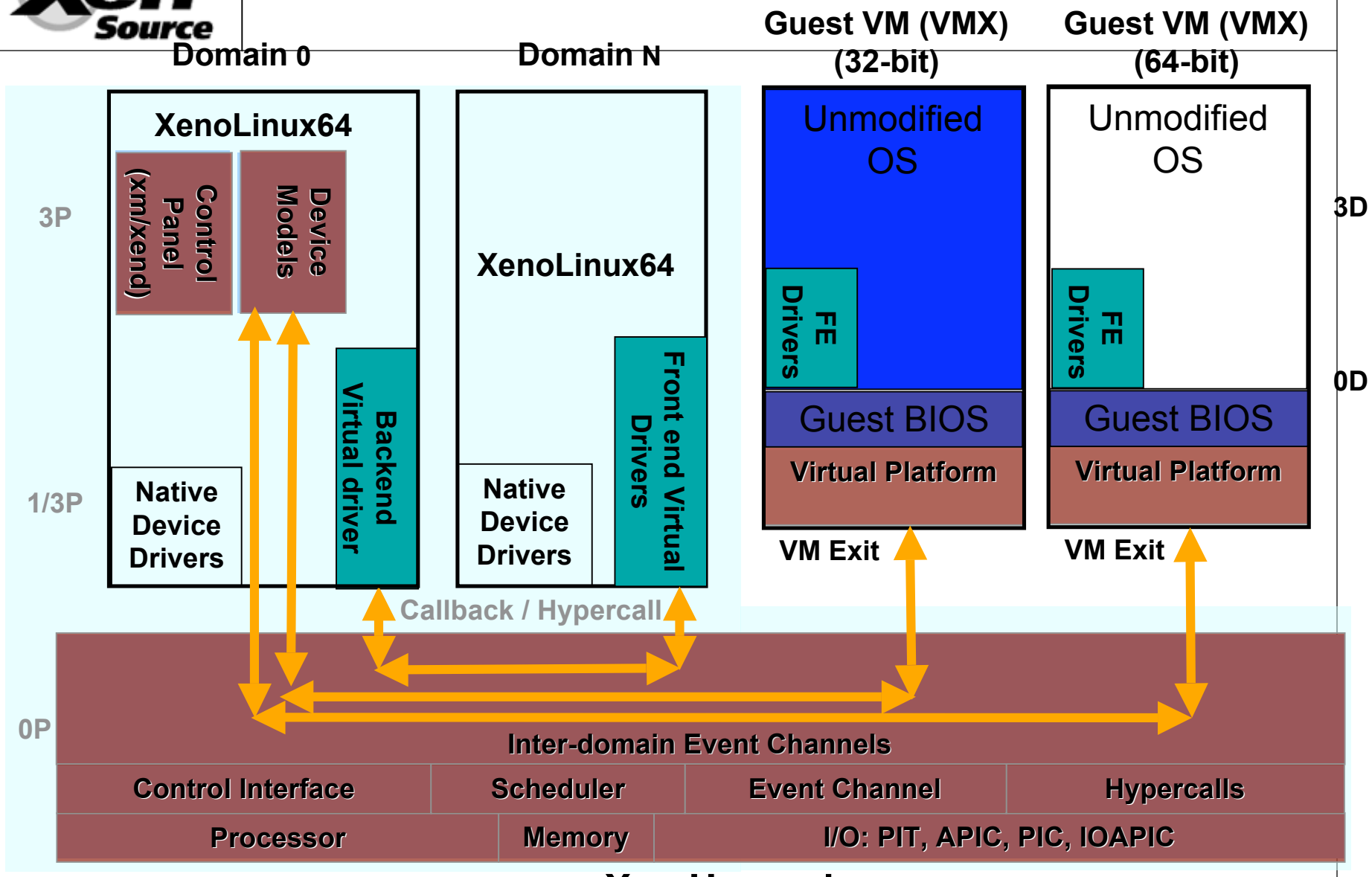
Writeable Page Tables : 4 – Re-hook



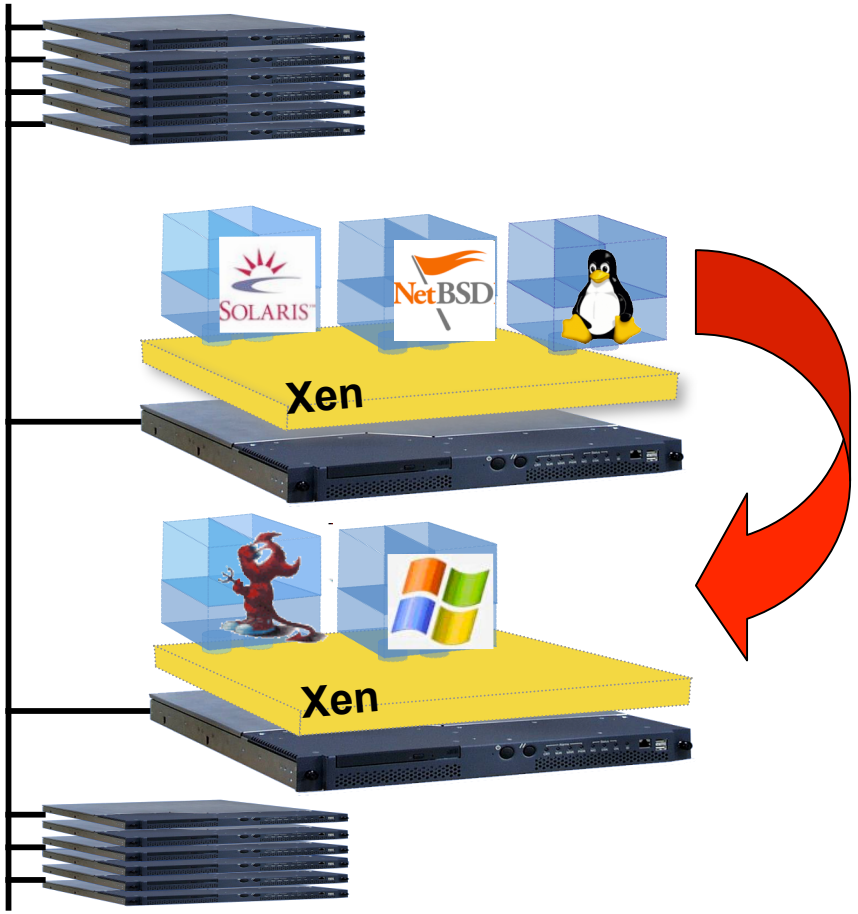
VT-x / Pacifica

- Enables Guest OSes to run without paravirtualization modifications
  - E.g. Windows XP/2003
- CPU provides traps for certain privileged instructions
- Shadow page tables used to provide MMU virtualization
- Xen provides simple platform emulation
  - BIOS, Ethernet (e100), IDE and SCSI emulation
- Install paravirtualized drivers after booting for high-performance IO

Xen with VT: Architecture Overview



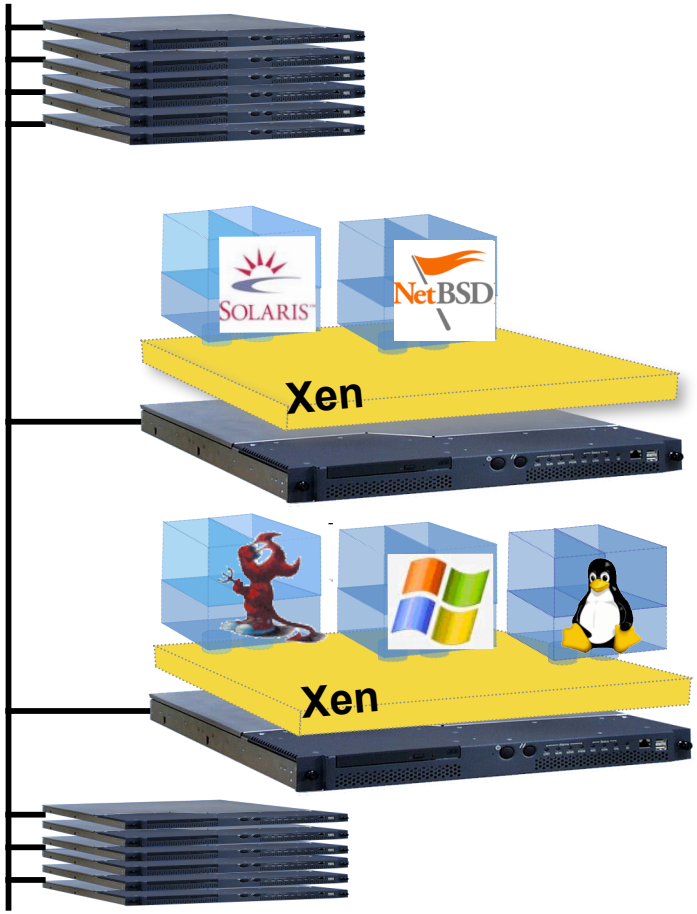
VM Relocation : Motivation



VM relocation enables:

- High-availability
  - Machine maintenance
- Load balancing
  - Statistical multiplexing gain

Some Assumptions



- Networked storage
  - NAS: NFS, CIFS
  - SAN: Fibre Channel
  - iSCSI, network block dev
  - drdb network RAID
- Good connectivity
  - common L2 network
  - L3 re-routeing

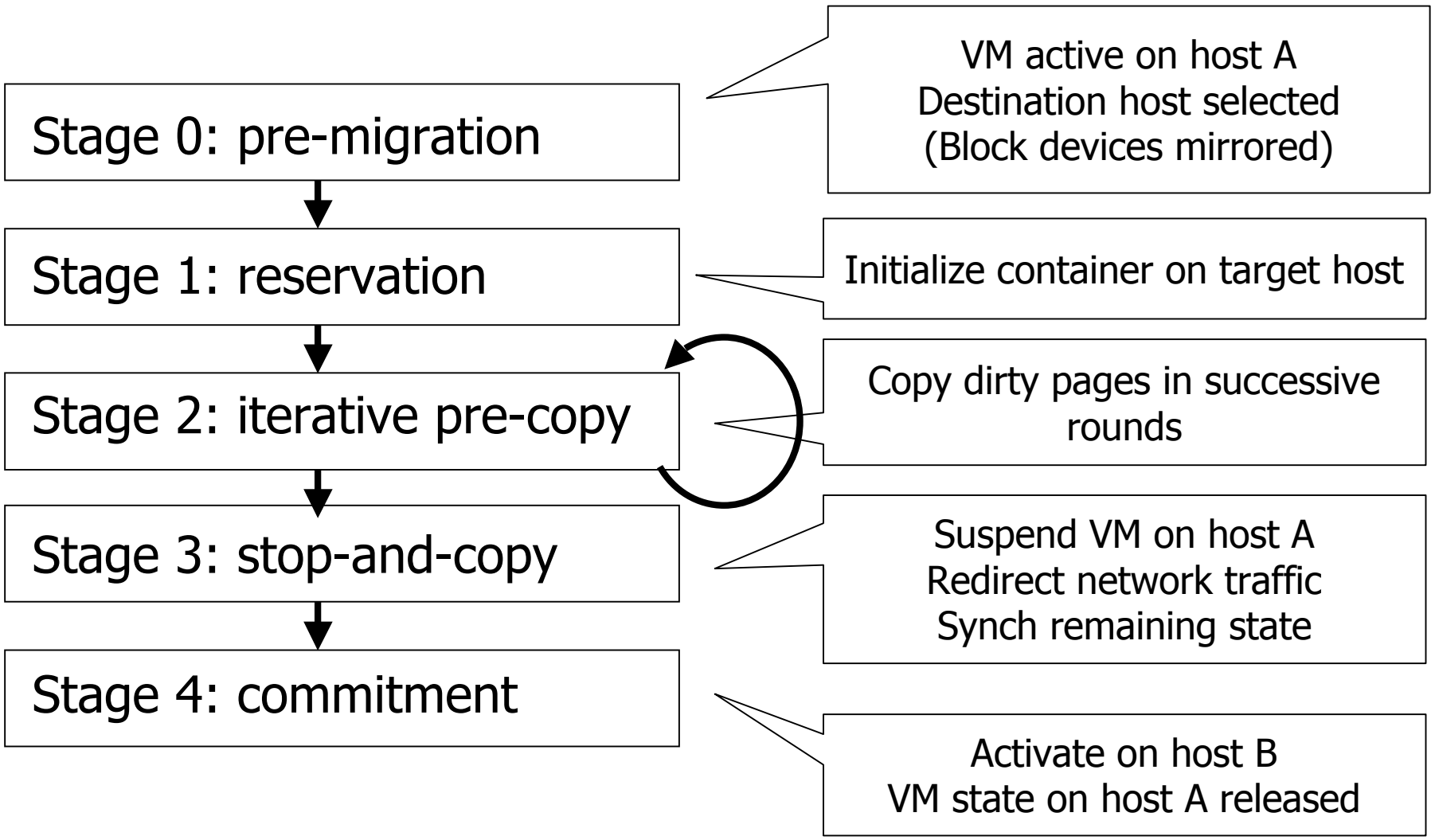
Challenges

- VMs have lots of state in memory
- Some VMs have soft real-time requirements
  - E.g. web servers, databases, game servers
  - May be members of a cluster quorum

  ➔ **Minimize down-time**
- Performing relocation requires resources

  ➔ **Bound and control resources used**

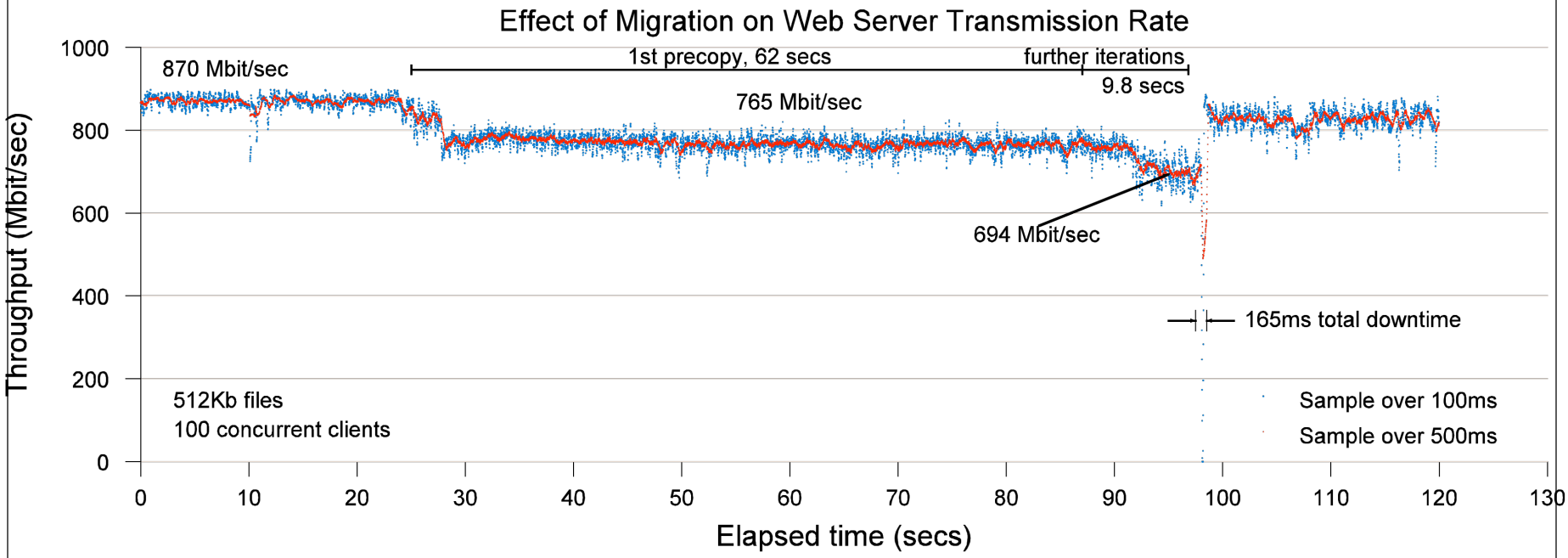
Relocation Strategy



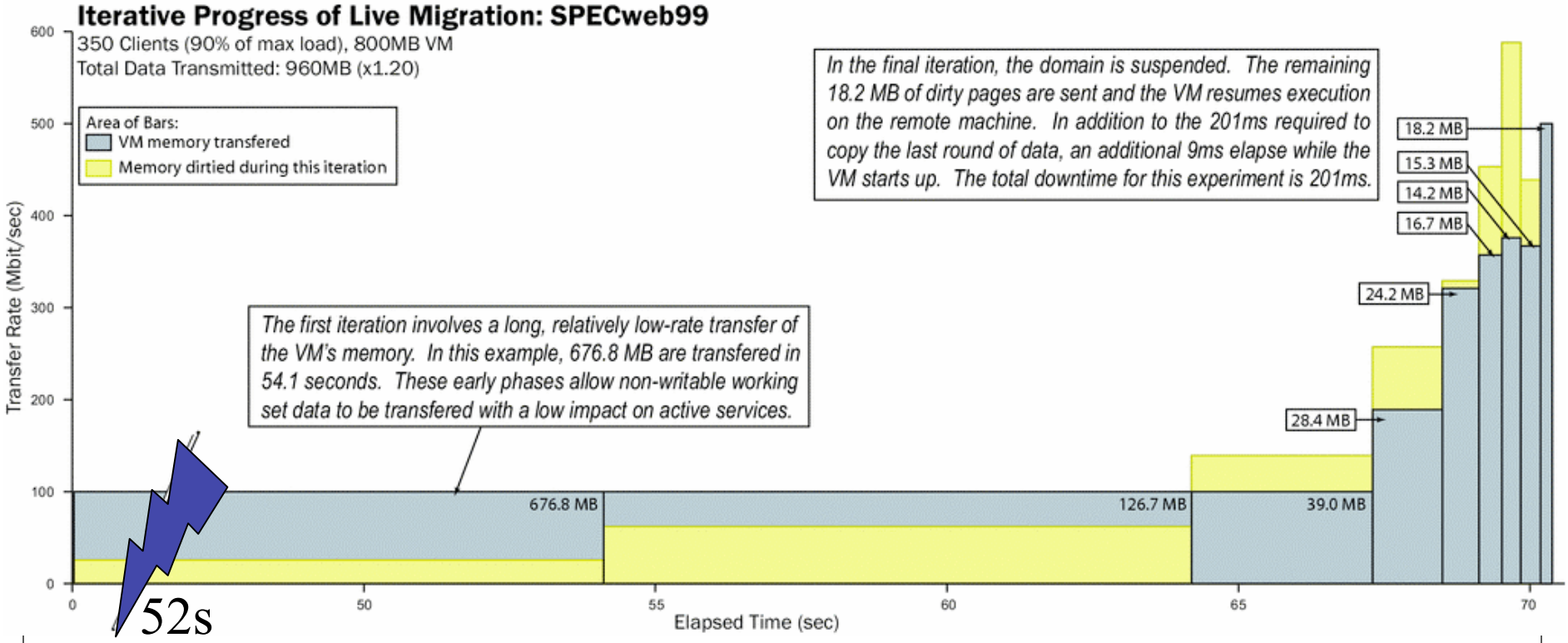
Rate Limited Relocation

- Dynamically adjust resources committed to performing page transfer
  - Dirty logging costs VM ~2-3%
  - CPU and network usage closely linked
- E.g. first copy iteration at 100Mb/s, then increase based on observed dirtying rate
  - Minimize impact of relocation on server while minimizing down-time

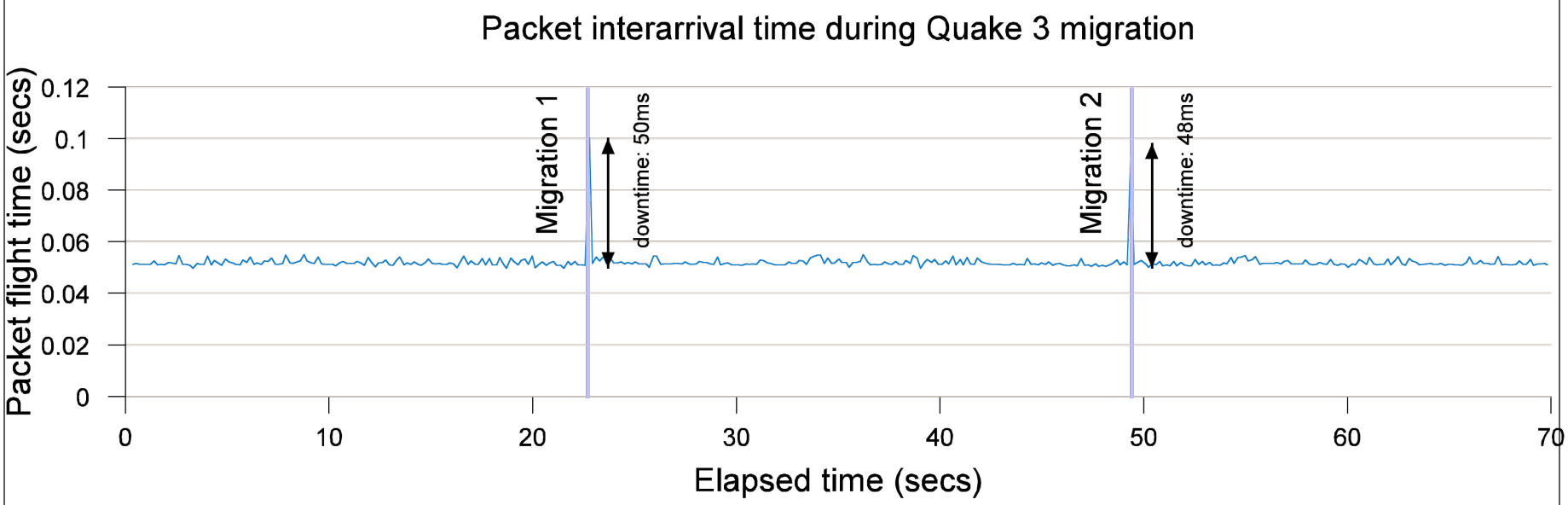
Web Server Relocation



Iterative Progress: SPECWeb



Quake 3 Server relocation



Thanks!

Contact information:

Simon Crosby

XenSource

(415) 819 1965

simon@xensource.com

www.xensource.com